CSE 5852, Modern Cryptography: Foundations Fall 2016

Lecture 21

Prof. Benjamin Fuller *Scribe: Lei Li*

1 Last Class

Signature Construction: Based on RSA-inv assumption we proposed the following signature scheme:

$Gen(1^n)$:

1. Choose two n-bit prime integers p and q by some special algorithm.
2. Compute $N = p \cdot q$ also compute parameter e, d.
3. output $vk = (e, N)$, $sk = (d, N)$

$Sign(d, N, m)$:

1. Compute $\sigma = m^d \mod N$

$Vfy(e, N, m, \sigma)$:

1. Compute $m' = \sigma^e \mod N$
2. output 1 if $m = m'$

The problem with such a scheme is it's not secure under chosen message attack. An attacker can choose a signature σ and compute a valid message $m = \sigma^e$. We proposed adding a function before applying d to the message, namely $\sigma = H(m)^d$. And the verify function should also be modified to check if $m' = H(m)$ and output correspondingly.

Hash Functions: The word "Hash" has different meanings to different people and here we define the hash function mapping the elements from the universe $\{0,1\}^*$ to the universe $\{0,1\}^n$. There are several properties we might want from such a Hash function:

1. One-way: This means easy to compute but very difficult to invert.
2. Collision Resistance: Be difficult to find two different messages m, m' and $H(m) = H(m')$.
3. Strong Universal: The definition can be found in previous lectures and informally, we want the outputs of such a hash function just like random (if the messages are chosen before knowing the description of the hash function.

2 Definitions of Collision Resistance

We first introduce the definition of collision Resistance for a hash function:

Definition *(Collision Resistance) A Hash function H is secured if for all PPTA there exists a negligible function ϵ such that $Pr[(m, m') \leftarrow \mathcal{A}, m \neq m' \text{ and } H(m) = H(m')] < \epsilon(n)$*

Having such a secured function is still not enough for our public signature. Because the public key in our scheme is known by public which the definition above didn't consider. Then we introduce the second definition:

Definition *(Collision Resistance) A family of Hash functions $\{H_i\}$ is secure if for all PPT A there exists a negligible function ϵ such that $\underset{i}{Pr}[(m, m') \leftarrow \mathcal{A}(i), m \neq m' \text{ and } H_i(m) = H_i(m')] < \epsilon(n)$*

The index i here is related to the public key, which being said, we can assume there is a hash family and we can select which one to use according to the key. However, the fact is no single hash function can be collision resistant. In practice, people just use one single hash function which has collisions, but very difficult to find those collisions. In fact, methods like SHA2, SHA3 (the current hash standards) are not keyed.

So far, the collision resistance is well defined, however, may be overcommitted, in practice, people don't care if there is collision for any kinds of input, but only care about if there is a collision for one specific input. So we introduce the definition of Second Preimage Resistance:

Definition *(Second Preimage Resistance) A Hash function H is secured if for all PPTA there exists a negligible function ϵ such that $\underset{i,m}{P}r[m' \leftarrow \mathcal{A}(i, m), m \neq m' \text{ and } H_i(m) = H_i(m')] < \epsilon(n)$*

Again, we can loose the definition a little bit because in practice, people actually don't care if the collision exist or not, but only care if the adversary can find it after seeing the output of the hash function. Then we introduce the definition of Preimage Resistance:

Definition *(Preimage Resistance) A Hash function H is secured if for all PPTA there exists a negligible function ϵ such that $\underset{i,m}{P}r[m' \leftarrow \mathcal{A}(i, H(m)) \text{ and } H_i(m) = H_i(m')] < \epsilon(n)$*

Please note in the definition of Preimage Resistance we no longer have $m \neq m'$. This is because the adversary only see $H(m)$, we are intrested in whether he can find a message whose mapping is $H(m)$, and it makes little sense to must find a different message.

3 Relationship Between Definitions

Overall, collision resistance infers second preimage resistance, and second preimage resistance infers preimage resistance if the hash function shrink a lot. To be specific, for a hash function $\{0,1\}^m \rightarrow \{0,1\}^n$, shrink a lot means m and n are not close.

Proof of collision resistance infers second preimage resistance: Use contradiction, first assume there exist $\mathcal{A}$ who takes input (i, m) and output m' such that $\underset{i,m}{P}r[m' \leftarrow \mathcal{A}(i, m), m \neq m'$

and $H_i(m) = H_i(m')] \geq 1/p(n)$ for some polynomial p. Then we build $\mathcal{A}'$ who output two different messages m, m' with same hash mapping:

1. Choose a random message m.
2. Call $\mathcal{A}$ on the chosen message m and receive m'.
3. Output m, m'.

This suggests the probability $\underset{i}{Pr}[(m, m') \leftarrow \mathcal{A}'(i), m \neq m' \text{ and } H_i(m) = H_i(m')] = \underset{i,m}{P} r[m' \leftarrow \mathcal{A}(i, m), m \neq m' \text{ and } H_i(m) = H_i(m')]$, which is inverse polynomial.

Proof of second preimage resistance infers preimage resistance: Assume H:$\{0,1\}^{2n} \rightarrow \{0,1\}^n$. Use contradiction, first assume there exist $\mathcal{A}$ who takes input (i, y) and output m' such that $\underset{i,m}{P} r[m' \leftarrow \mathcal{A}(i, y) \text{ and } y = H_i(m')] \geq 1/p(n)$ for some polynomial p. Then we build $\mathcal{A}'$ who takes input i, m and output a messages m':

1. $\mathcal{A}'$ takes input i, m
2. Compute $y = H_i(m)$
3. Call $\mathcal{A}$ on i, y and receive m'.
4. Output m'.

Then we need to show $m' \neq m$ a significant fraction of the time (we'll some that $m' \neq m$ half the time). Define the event $I_k(m) = 1$ if there is only one preimage for message m. The obvious fact is there are at most $2^n - 1$ messages can have only one preimage, where n is the bit length of hash function. We assume the bit length of input is twice the large of hash table, then the probability for a message to have only one preimage is $Pr[I_k(m) = 1]$ is at most $(2^n - 1)/2^{2n}$ which is approximately $1/2^n$. Now we know $\underset{i,m}{P} r[m' \leftarrow \mathcal{A}(i, y) \text{ and } y = H_i(m')] \geq 1/p(n)$, divide the equation into three cases:

$$\underset{i,m}{P} r[m' \leftarrow \mathcal{A}(i, y) \text{ and } y = H_i(m') | I_k(m) = 1] \Pr[I_k(m) = 1] +$$

$$\underset{i,m}{P} r[m' \leftarrow \mathcal{A}(i, y) \text{ and } y = H_i(m') | I_k(m) = 0 \cap m = m'] \Pr[I_k(m) = 0 \cap m = m'] +$$

$$\underset{i,m}{P} r[m' \leftarrow \mathcal{A}(i, y) \text{ and } y = H_i(m') | I_k(m) = 0 \cap m \neq m'] I_k(m) = 0 \cap m \neq m' \geq 1/p(n)$$

$\Pr[I_k(m) = 1] \leq 1/2^n$ so the first line is at most $1/2^n$. Then we claim the frequency of the third case is no less than the second case, and they are equal only when m has two preimages. Finally we have $\underset{i,m}{P} r[m' \leftarrow \mathcal{A}(i, y) \text{ and } y = H_i(m') | I_k(m) = 0 \cap m \neq m'] \geq (1/p(n) - 1/2^n)/2$, which is an inverse polynomial. This suggests $\mathcal{A}'$ breaks second preimage resistance.

In the proof above, consider another hash function H:$\{0,1\}^{n+1} \rightarrow \{0,1\}^n$, then the probability for a message to have only one preimage will be approximately $1/2$, and the whole proof will not work. This is why we want the hash function to shrink a lot to let most messages have at least two preimages.

4 Random Oracle Model

Collision resistance may not be enough. It is obviously not secure if there are many collisions in H but its not clear that the adversary needs to find collisions to break RSA. Consider RSA-inv:

$\sigma_1 = H(m_1)^d$, $\sigma_2 = H(m_2)^d$, it may be possible for an adversary to compute a hash function for $m_1 \cdot m_2$ if they can predict $\sigma_1\sigma_2 = (H(m_1)H(m_2))^d$. So we need some kind of multiplication resistance of the hash function.

Then we introduce the Random Oracle Model [BR93]. This is the strongest possible requirement of a hash function. The hash function is assumed to be a black box that implements a random function which everyone can get access to. Each participant only hash input and output access and doesn't know how the function works. The black box should act as a random function but it is deterministic for each input. We know we don't have any real random functions. Furthermore, we know for hash functions we don't ask a black box to compute our hash we do it locally on our computer. We will use this Random Oracle Model to prove security and then try to replace it with a real hash function and hope it works. The properties we want from such a black box is:

1. Nobody can understand its code.
2. If x has not been queried then $H(x)$ should be random.

References

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.